УДК 004.4

ПРИМЕНЕНИЕ STEM-ТЕХНОЛОГИЙ В СРЕДНЕМ ПРОФЕССИОНАЛЬНОМ ОБРАЗОВАНИИ

М.В. Рожкова[1]

1 *ros.seminar@bk.ru*; Краснодарский колледж управления, техники и технологий

Изложен опыт комплексного изучения математики, с использованием современных информационных технологий, в техническом колледже.

Ключевые слова: математика, информационные технологии, компьютерная алгебра.

Революционные изменения в жизни общества ставят новые, трудные проблемы перед системой образования.

Как объять необъятное, соединить специализацию с универсальностью, максимально сократить путь для начинающих к передовым рубежам науки, соблюсти разумное соотношение между наглядностью и научной точностью и т.д.

Особенно актуально это в системе среднего профессионального образования, где наряду с получением общего образования, учащиеся получают и профессию.

1. Американский проект STEM, взгляд из России

Признавая, на государственном уровне, недостатки математического и технического образования в своей стране, США разработали проект STEM.

STEM — Science, Technology, Engineering, and Mathematics, инициатива в области образования, которая начала осуществляться в США с лета 2013 г.

Проект рассчитан на 5 лет. На его выполнение выделено 15 млрд. $. В процессе его осуществления планируется переподготовить около 100 тыс. преподавателей учебных заведений, в основном колледжей.

В проекте реализуется старая как мир идея — связь образования с жизнью и производством. Официальный обзор проекта изложен в [5].

Некоторые страны, например, Казахстан, тоже реализуют эту идею на государственном уровне, но в своем прочтении — STEAM — добавлено направление Art — искусство.

В России, а ранее в СССР, всегда была популярна тема связи науки и образования, образования и производства, а теперь образования и современных информационных технологий.

Смысл нашей стратегии [1] не в том, чтобы студенты и школьники, а проект доступен и старшеклассникам, освоили одну из операционных систем, могли писать программы на одном из языков программирования, имели навыки работы с одной из систем компьютерной алгебры. Хотя это и дополнительный плюс. Хорошая основа для профессии системного администратора, программиста или специалиста в области прикладной математики.

Предложена некая интеллектуальная основа, технологическое дерево, которое может дать плоды в самых различных областях науки и техники, где нужны и важны вычислительные подходы.

Смысл и свехрзадача — это комплексное освоение всей триады IT-технологий: системная среда - программное окружение - технический инструментарий.

Человек усвоил знания, если они стали частью его натуры, проникли в подсознание. Поэтому изучение научных понятий желательно сопровождать их компьютерной реализацией, которая служит и иллюстрацией абстрактной теории и, попутно, позволяет выработать, на подсознательном уровне, крайне полезные навыки в области IT-технологий.

2. Выбор технических средств

Информационные технологии давно перестали быть уделом профессионалов. ЭВМ сейчас такой же бытовой прибор как чайник и холодильник. Это часть нашей повседневной жизни. Сейчас овладение информационными технологиями не доблесть, а жестокая необходимость.

Важный момент при выборе IT-инструментария его ориентированность на обучение, научный подход к овладению будущей специальностью.

Другой немаловажный параметр - цена решения и качество используемого продукта.

а) *Выбор операционной системы*

Из сотен операционных систем линейки Linux выбор был однозначно сделан в пользу Linux Debian, имеющий наиболее дружественный интерфейс среди многих других систем на открытом коде.

Операционную систему Linux Debian создали в 1993 г. Debra Lynn и Ian Murdock. Это один из старейших дистрибутивов Linux.

Нами используется один из клонов Debian, который установлен в тысячах школ и колледжей по всему миру. Речь идет о Debian Edu - Skolelinux — он предоставляет готовое окружение полностью настроенной школьной сети.

В дистрибутив Debian входит более 43 000 пакетов. Одних только программ для математических вычислений более 300.

В их числе Axiom, Cadabra, Cantor, Euler, GAP, KAlgebra, Mathomatic, Maxima, Octave, Pari-GP, Qalc, Relational, Sagemath.

Представление о работе с Debian можно получить ознакомившись с книгой [2].

б) *Выбор базового языка программирования*

Как бы идеально не была устроена операционная система ее использование под разные задачи требует отдельной настройки. С компьютером нужно "разговаривать"и этим языком общения является язык программирования.

Языков программирования очень много - их тысячи! Они имеют разный функционал и назначение. Низкоуровневые, специальные, универсальные.

Для нужд образования, более всего, подходят языки универсальные.

И кажется есть очевидное решение — это языки семейства C/C++, Visual C и т.д. Однако эти языки рассчитаны на профессиональных программистов. Множество тонкостей этих суперуниверсальных языков не нужны человеку просто изучающему математику. Их возможности просто избыточны.

Язык математических вычислений Fortran великолено справляется с инженерными задачами, но не очень удобен в символьных вычислениях и при создании интерфейса программного продукта, который предполается использовать в учебных

целях.

После длительных изысканий наш выбор был остановлен на языке Python.

Python — высокоуровневый, объектно-ориентированный, исполняемый язык программирования общего назначения, с динамической типизацией, создан голландцем Guido van Rossum в 1991 г.

Python позицируется как язык программирования, доступный для всех. Его синтаксис минималистичен, не формален, ориентирован на начинающих программистов.

Python имеет много различных вариаций и расширений — RPython, CPython, Cython, Jython и т.д. и поэтому применяется практически во всех сферах IT-технологий.

На Python осуществляются тысячи проектов, в которых заняты миллионы профессиональных программистов.

Фирма Yandex язык Python использует как базовый для обучения школьников.

Немаловажно, что Python распространяется по лицензии на свободное программное обеспечение.

Хороший обзор линейки Python 3.x дан в книге [3].

в) *Выбор системы компьютерной алгебры*

Средств для научного анализа и математических вычислений великое множество — MatLab, MathCad, Maple, Magma и т.д., но все они весьма не дешевы в приобретении и владении.

Есть сотни бесплатных проектов на открытом коде. Часть из них перечислена при описании пакетов в ОС Debian.

Но главная проблема не в платности и бесплатности, а в том как пользоваться этими пакетами компьютерной алгебры.

В популярном у специалистов по дискретной математике пакете GAP — Groups, Algorithms, Programming, более 4000 встроенных функций, вычисляющих те или иные числовые или символьные характеристики алгебраических объектов.

К сожалению, кнопок для вызова конкретных функций программисты могут создать только конечное число, а математических задач бесконечно много!

Поэтому каждый программный комплекс имеет свой внутренний, встроенный, язык программирования. Но все эти встроенные языки разные. И перейдя на другой пакет придется переучиваться.

Как обойти эту проблему. В 2005 г. профессор В. Штайн из Вашингтонского университета (США) предложил новую оригинальную идею. Он не стал создавать новый программный комплекс с нуля. В. Штайн создал проект на открытом коде, взяв в качестве встроенного языка программирования Python. Кроме того, включил в проект уже существующие 90 пакетов на открытом коде, в том числе: NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R.

Проект имеет реализацию в виде Live USB Key на основе Linux Debian. Поэтому не нужно устанавливать Linux, а достаточно создать флешку, и с нее работать в Sage на любом компьютере.

Проект активно развивается. Каждый квартал выходит новая версия. Текущий релиз Sage 7.4.

Проект Sage подробно описан в книге [4].

3. Математика и программирование на практике

Первоначально все задумывалось с достаточно прозаической целью — как источник для написания рефератов и кружковой работы.

По мере реализации проекта стало ясно, что технология может использоваться для создания и разработки комплексов программ как в области прикладного программирования так и проведения научных исследований учащимися.

Преподавателям хорошо известно, что некоторые учащиеся не просто не знают, не любят, а боятся математики как огня и нечистой силы — скучная, формальная, заумная.

Однако есть море задач и математических загадок, которые и понятны, и забавны, и нетривиальны. Стоит только ознакомиться, например, с книгой [7].

Возьмем для примера число 666. Это "нехорошее число"даже вынесено в заголовок книги [7].

Однако мы используем другое истолкование этого числа, отличное от толкования в книге [7].

Определение. *Функция Эйлера $\varphi(n)$ от натурального числа n — это количество натуральных чисел, меньших n и не имеющих с n общих делителей.*

Число 666, среди прочего, интересно тем, что $\varphi(666) = 216 = 6 \cdot 6 \cdot 6$.

Назовем число *нехорошим*, если функция Эйлера от него равна произведению цифр в его десятичной записи.

Не очень понятно есть ли еще такие числа, кроме 666. А если есть, то может ли их число быть бесконечным.

Задача вполне серьезная и интересная. Однако до всяких теоретических исследований лучше всего провести численный эксперимент, написав небольшую программу в Sage или GAP. Приводим ее ниже

```
Prod:=function(x)
local a,b,y;
     y:=1;
b:=x; a:=b mod 10;
     y:=y*a;
repeat
       b:=Int(b/10); a:=b mod 10; y:=y*a;
until b<10;
       return(y);
end;

Euler:=function(m,n)
local i,j,k,L;
L:=[];
i:=m;
while i<n do
      j:=Prod(i);
    if j>0 then
       k:=Phi(i);
           if j=k then
              Print(i," ","->"," ",k, "\n"); Add(L,i);
           fi;
    fi;
```

```
            i:=i+1;
od;
   return(L);
end;
```

Запускаем программу с параметрами $Euler(1,10^6)$, т.е. проверяем на *нехорошесть* все числа до миллиона. И получаем результат

```
[24,26,87,168,388,594,666,1998,2688,5698,5978,6786,7888,68796]
```

Найдено ровно 14 чисел. Как остроумно заметили учащиеся: “Черт и его чертова дюжина!” Забавно, что “нехорошим числом” оказалось и число 1998 — год, когда в России случился дефолт.

Можно провести вычисления до миллиарда и до триллиона. Ничего нового мы не получим. Теперь осталось теоретически доказать, что других “нехороших” чисел нет! Когда ты уверен в результате это сделать гораздо проще, чем пребывая в неведении.

Какие еще проекты научного и учебного направления можно реализовывать в рамках выбранной нами программной платформы?

На момент написания статьи в Краснодарском колледже управления, техники и технологий представлено примерно 50 тем, связанных с компьютерной алгеброй и теорией чисел, доступных для учащихся в рамках кружковой и научной работы.

Вот одна из предложенных задач.

Это не решенная до сих пор проблема Коллатца. Сформулирована она была в 1937 г., т.е. ей скоро исполнится 80 лет! Это серьезный вызов всей математической общественности планеты Земля.

Формулировка гипотезы Коллатца. *Берем натуральное число n. Если оно четное, то делим его на 2 до тех пор, пока оно не станет нечетным числом m. Заменяем n на $3m+1$. Полученное число опять делим на 2 и т.д. Доказать, что каково бы не было исходное число n в итоге всегда получится 1.*

Пример. Возьмем $n = 9$. Вычисляем, отбрасывая четные члены,

$$9 \to 7 \to 11 \to 17 \to 13 \to 5 \to 1.$$

Формулировка проблемы Коллатца проста и легко поддается программированию. Например, приведенная ниже программа для каждого из чисел k, принадлежащих промежутку $[n, m]$, вычисляет через сколько итераций число $2k+1$ превратится в 1.

```
Tr:=function(a)
local x,y;
x:=a;
repeat x:=x/2;
until x mod 2 =1;
y:=3*x+1;
return(y);
end;

Collatz:=function(m,n)
local i,j,k,t,L;
L:=[];
```

```
for i in [m..n] do
                  j:=6*i+4; t:=0;
        repeat j:=Tr(j); t:=t+1;
        until j =4;  Print(2*i+1,"->",t,"\n");
Add(L,t);
i:=i+1;
od;
return(L);
end;
```

Запустим программу с параметрами $Collatz(987, 1008)$ и получим

```
1975->10      1977->51      1979->51
1981->34      1983->15      1985->15
1987->32      1989->5
1991->15      1993->15      1995->15
1997->15      1999->15      2001->15
2003->51      2005->39      2007->12
2009->5       2011->12      2013->22
2015->32      2017->22
```

Удивительное и очень редкое явление — шесть подряд нечетных чисел от 1991 до 2001 переходят в 1 ровно за 15 итераций. Много доступных для школьников задач по теории чисел и криптографии можно найти в [6].

Подобные задачи знакомят учащихся с трудными математическими проблемами, которые легко можно понять! Более того, им по силам начинать их решать, проводя компьютерные эксперименты. И очень даже может быть, что проводя эти эксперименты, мы не только привьем им любовь к математике, но и дадим им возможность решить реальную проблему. Внести свой вклад науку уже в столь юном возрасте!

Литература

1. Рожков А. В. Преподавание математики и информатики в ведущих университетах мира и опыт КубГУ / А. В. Рожков, М. В. Рожкова // Университеты в системе поиска и поддержки математически одаренных детей и молодежи: материалы I Всероссийской научно-практической конференции. - Майкоп, 2015. - С. 116-121.

2. Негус К. Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд / К. Негус, Ф. Каэн. — СПб.: Питер, 2011. — 352 с.

3. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. - СПб.: Символ-Плюс, 2009. - 608 с.

4. Finch C. Sage Beginner's Guide / C. Finch. - Packt Publishing, 2011. - 366 p.

5. Federal Science, Technology, Engineering, and Mathematics (STEM) Education, 5-year strategic plan. A Report from the Committee on STEM Education National Science and Technology Council, May 31, 2013.

6. Рожков А. В. Теоретико-числовые методы в криптографии: учебное пособие / А.В. Рожков, О.В. Ниссенбаум. - Тюмень: Издательство Тюменского государственного университета, 2007. - 160 с.

7. Л. Г. Сид Мир математики. Замечательные числа. Ноль, 666 и другие бестии / Ламберто Гарсия дель Сид. Перев. с исп. - М.: Де Агостини, 2014. - 160 с.

APPLICATION OF STEM TECHNOLOGIES ON AVERAGE PROFESSIONAL EDUCATION

M.V. Rozhkova

Experience of complex studying of mathematics in technical college with use of modern information technologies is stated.

Keywords: mathematics, information technologies, computer algebra.

УДК 5530.12+531.51

ТЕХНОЛОГИЯ РЕШЕНИЯ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ НА ОСНОВЕ СИНТАКСИЧЕСКОГО АНАЛИЗА ЕЕ ТЕКСТА

В.Э. Садриев[1]

1 *vsadriew@mail.ru*; Казанский (Приволжский) федеральный университет

Описана технология преобразования синтаксических конструкций естественного языка в соответствующие им конструкции на языке программирования.

Ключевые слова: программирование,естественный язык, задача, синтаксический анализ.

Как правило, текст подавляющего числа задач, в том числе и по программированию, строится с учётом всех правил грамматики и синтаксиса русского языка. Компьютерная программа - это тоже текст, но составленный по правилам некоторого языка программирования, который также имеет свою чёткую структуру, грамматику, правила. Таким образом, возникает вопрос, возможно ли найти соответствие грамматико-синтаксических конструкций естественного языка, каким является русский, соответствующие им конструкции в формальных языках, к которым относятся языки программирования.

При этом очевидно, что данные предположения могут быть справедливы для относительно несложных конструкций, потому что количество информации, заключенной в отдельные слова, сравнительно невелико, поэтому и соответствующие им конструкции в языках программирования также будут обладать сопоставимым относительно малым информационным объёмом.

Здесь предполагается провести сравнительный анализ языковых конструкций и проследить, каким именно образом будет изменяться синтез эквивалентных им участков кода программ. Здесь и далее под понятием «язык» будет подразумеваться текст на естественном языке, в первую очередь русском, а под понятием «код» будет предполагаться текст, созданный на одном из языков программирования.

Идея данного исследования возникла как результат преподавания курса программирования в школе в рамках уроков информатики. Основная проблема заключается в том, что часто учащиеся «не видят» в тексте задачи ни то, что дано, ни то, что нужно найти. Поэтому при разборе условия зачастую приходится акцентировать внимание на отдельных словах, словосочетаниях, а то и целых оборотах.

В стандартном предложении русского языка используются следующие основные грамматические конструкции: подлежащее, сказуемое, дополнение, определение, обстоятельство, причастный оборот, деепричастный оборот, остальные вводные конструкции. При составлении предложений обычно отталкиваются от подлежащего со сказуемым, к которым по определённым правилам добавляются остальные составляющие.

В общем случае код, записанный разными языками программирования, выполняющий одни и те же действия, внешне может быть записан с использованием различных технологий, кардинально между собой отличающимися. В рамках данной статьи будем оттал-